



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение высшего  
образования

**"МИРЭА - Российский технологический университет"**

**РТУ МИРЭА**

---

Институт информационных технологий (ИТ)  
Кафедра математического обеспечения и стандартизации информационных  
технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №8\_2**  
**по дисциплине**  
**«Структуры и алгоритмы обработки данных»**

Тема. Алгоритмические стратегии или методы разработки алгоритмов

Выполнил студент группы ИКБО-42-23

Голев С.С.

Принял ассистент

Муравьёва Е.А.

Москва 2024

## ОГЛАВЛЕНИЕ

ЦЕЛЬ РАБОТЫ.....	3
УСЛОВИЯ ЗАДАЧ.....	4
1. ОТЧЁТ ПО ЗАДАНИЮ 2 .....	5
1.1. Код используемый в программе.....	5
1.3. Результаты тестирования.....	6
4.ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ .....	7

## **ЦЕЛЬ РАБОТЫ**

Получить навыки применения методов, позволяющих сократить число переборов в задачах, которые могут быть решены только методом перебора всех возможных вариантов решения.

## УСЛОВИЯ ЗАДАЧ

### Задание 1:


Устная часть.

### Задание 2:

Разработать алгоритм решения задачи с применением метода, указанного в варианте и реализовать программу.

Оценить количество переборов при решении задачи стратегией «в лоб»-грубой силы. Сравнить с числом переборов при применении метода.

Оформить отчет в соответствии с требованиями.

<p>Треугольник имеет вид, представленный на рисунке. Напишите программу, которая вычисляет наибольшую сумму чисел, расположенных на пути, начинающемся в верхней точке треугольника и заканчивающегося на основании треугольника.</p>  <pre>      7      3 8     8 1 0    2 7 4 4   1 5 2 6 5</pre>	<p>Динамическое программирование</p>
---	--------------------------------------

# 1. ОТЧЁТ ПО ЗАДАНИЮ 2

## 1.1. Код используемый в программе

Опишем функции, используемые в программе для решения задач.

```
int max_sum(std::vector<std::vector<int>>& triangle)
{
    int n = triangle.size();

    std::vector<int> maxSums = triangle[n - 1];

    for (int i = n - 2; i >= 0; --i)
        for (int j = 0; j <= i; ++j)
        {
            lenDyn++;
            maxSums[j] = triangle[i][j] + std::max(maxSums[j], maxSums[j + 1]);
        }
    return maxSums[0];
}
```

*Рисунок 1.1 – Функция реализованная методом динамического программирования*

```
int brute_force(const std::vector<std::vector<int>>& triangle, int row, int col)
{
    if (row == triangle.size() - 1)
        return triangle[row][col];

    lenBru++;
    int leftPathSum = brute_force(triangle, row + 1, col);
    int rightPathSum = brute_force(triangle, row + 1, col + 1);

    return triangle[row][col] + std::max(leftPathSum, rightPathSum);
}
```

*Рисунок 1.2 – Функция реализованная методом грубой силы*

```
int main()
{
    setlocale(LC_ALL, "RUS");

    std::vector<std::vector<int>> vec = { { 7 }, {3,8}, {8,1,0}, {2,7,4,4}, {4,5,2,6,5} };
    int result = max_sum(vec);
    std::cout << "Наибольшая сумма на пути: " << result << ' ' << lenDyn << std::endl;
    std::cout << "Наибольшая сумма на пути (алгоритм грубой силы): " << brute_force(vec, 0, 0) << ' '
    << lenBru << std::endl;
}
```

*Рисунок 1.3 – Основная функция программы*

### 1.3. Результаты тестирования

Протестируем функции, используемые во втором задании.

```
Наибольшая сумма на пути: 30 | Количество переборов: 10  
Наибольшая сумма на пути (метод грубой силы): 30 | Количество переборов: 15
```

*Рисунок 2.4 – Тестирование программы*

#### **4.ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ**

1. Лекции по Структуры и алгоритмы обработки данных / Рысин М. Л.  
Москва, МИРЭА — Российский технологический университет.
2. Материалы по дисциплине Структуры и алгоритмы обработки данных /  
Скворцова Л. А. Москва, МИРЭА — Российский технологический университет.